Practical Reliable Bayesian Recognition of 2D and 3D Objects Using Implicit Polynomials and Algebraic Invariants

Jayashree Subrahmonia, David B. Cooper, and Daniel Keren

Abstract—Patches of quadric curves and surfaces such as spheres, planes, and cylinders have found widespread use in modeling and recognition of objects of interest in computer vision. In this paper, we treat use of more complex higher degree polynomial curves and surfaces of degree higher than 2, which have many desirable properties for object recognition and position estimation, and attack the instability problem arising in their use with partial and noisy data. The scenario discussed in this paper is one where we have a set of objects that are modeled as implicit polynomial functions, or a set of representations of classes of objects with each object in a class modeled as an implicit polynomial function, stored in the database. Then, given partial data from one of the objects, we want to recognize the object (or the object class) or collect more data in order to get better parameter estimates for more reliable recognition. Two problems arising in this scenario are discussed in this paper: 1) the problem of recognizing these polynomials by comparing them in terms of their coefficients, which are global descriptors, or in terms of algebraic invariants, i.e., functions of the polynomial coefficients that are independent of translations, rotations, and general linear transformation of the data; and 2) the problem of where to collect data so as to improve the parameter estimates as quickly as possible. We solve these problems by formulating them within a probabilistic framework. We use an asymptotic Bayesian approximation which results in computationally attractive solutions to the two problems. Among the key ideas discussed in this paper are the intrinsic dimensionality of polynomials and the use of the Mahalanobis distance as an effective tool for comparing polynomials in terms of their coefficients or algebraic invariants.

Index Terms—Implicit polynomials, algebraic invariants, Bayesian recognition, Mahalanobis distance.

1 Introduction

In this paper, an entirely new approach is presented to low computation cost, highly accurate recognition of complex 2D and 3D objects in arbitrary position when some data may be missing due to occlusion or incorrect segmentation. The system described is a unified optimal system within a Bayesian decision-theoretic framework.

Implicit polynomial functions in x, y, z (or in x, y for curves in images) of degree greater than 2 can represent connected and disconnected objects in 3D (or 2D) space that are considerably more complicated than can be represented by quadrics (e.g, spheres, cylinders, cones, etc.) or by superquadrics. These polynomials, which are the focus of this paper, have great potential for object recognition and position estimation. Most of the early work on implicit polynomial curves and surfaces was limited to quadrics, thus dealing with representations that had modest expressive power. The fitting algorithms however were simple, the computational cost small and the resulting polynomial coefficients were stable [7], [4], [18], [16], [2], [17] Implicit polynomials of degree greater than 2, on the other hand, have great modeling power for complicated objects and can be made to fit to data very well, but their coefficients may be sensitive to small changes in the data. This poses a problem for object recognition because one would like to compare curves and surfaces based on their polynomial coefficients, or functions of the coefficients that represent only shape, i.e., that are invariant to translation and rotation, or any general linear transformation. In this paper, the problem of coefficient instability is addressed by looking at the intrinsic dimensionality of the polynomials and providing an appropriate distance metric for comparing two sets of coefficients or two sets of invariants. Finally, the problem of how to collect more data efficiently so as to improve the parameter estimates as quickly as possible is addressed.

2 Overview of the Recognition Approach

An implicit polynomial function representation is a curve in 2D (or surface in 3D) which is the *zero set*, i.e., the set of points (x, y) in 2D (or (x, y, z) in 3D) for which the polynomial function $f(x, y)$ on 2D (or $f(x, y, z)$ on 3D) is zero. For example, a circle in the x, y plane with center at the origin and radius R is the set of points which satisfy the equation $x^2 + y^2 - R^2 = 0$.

The fundamental 2D or 3D object recognition problem is that there are L stored object models in a data base. These models are 2D curves for object boundaries (internal and

- *J. Subrahmonia is with the Handwriting Algorithms Group at the IBM T.J. Watson Research Center. E-mail: jays@watson.ibm.com.*
- *D.B. Cooper is with the Laboratory for Engineering Man/Machine Systems, Division of Engineering, Brown University.*
- *D. Keren is with the Department of Mathematics and Computer Science, University of Haifa, Israel.*

Manuscript received Aug. 31, 1994, revised Sept. 14, 1995.
Recommended for acceptance by S.K. Nayar.
For information on obtaining reprints of this article, please send e-mail to: transactions@computer.org, and reference IEEECS Log Number P95183.

external) in images, and 3D object boundaries in range space. Each is stored in some *standard* position. The system has data over a portion of the object boundary. The system has to determine which of the L stored models is present in the data. A generalization of this problem is that the system stores L descriptions, one for each class of 2D (or 3D) objects. The system has measurements over a portion of one object in arbitrary position and must make a minimum probability of error decision as to which of the L classes the measured object belongs. The liberty that we are taking in this work is in assuming that the total data set has been segmented such that the data subset that our system must recognize is associated with a single object in the data base. For 3D range data, the segmentation is often relatively easy to do. For 2D image data, the segmentation is often a major part of the object recognition problem. We feel that in the more complicated situations, the segmentation would either be done with basic existing methods such as [20], [8], [19], or with a partially model-based approach which would involve using some of the model recognition methodology from the present paper.

The direct approach to the solution of the recognition of the L stored object recognition problem is to take each of the L stored models, transform it (i.e., translate it and either rotate it or linearly transform it, whichever is appropriate) in order to align it with the data as well as possible, and then check for the sum of squared distances from the data points to each of the aligned models. The model for which this sum of squared distances is a minimum is the recognizer's choice. The drawback of this approach is, first, the data is processed L times, and second, each model must be aligned with the data. The processing required is huge. The computational requirements of our approach, described in the following paragraph, will often be orders of magnitude less.

Our approach is to fit an implicit polynomial to the data. Then a few invariants are computed for the coefficients of the polynomial. These are functions of the coefficients that are invariant to translations and either rotations or linear transformations, whichever is appropriate, of the data. Typically, we use five to seven invariants for implicit polynomials for 3D surfaces or 2D curves. Then a weighted distance measure is used for comparing the vector of invariants for the polynomial fit to the data to be recognized with a vector of invariants for each of the L stored polynomials in the data base. This recognizer is the minimum probability of error recognizer for recognition based on the invariants, i.e., *it is impossible to find a more reliable recognizer.* The recognizer performs well even when data is missing from portions of the object. The required computation is small for reasons that will become clear as details of the recognizer are presented in the paper. Note, among these reasons is that simple *explicit* expressions are now available for the approximate distance from a data point to an implicit polynomial curve or surface (see Appendix A). This is not the case with explicit representations such as B splines where the computation of the distance from a point to the curve or surface is much higher because there is not an explicit expression for the distance and nonlinear iterative algorithms must be used for the computation.

Effective use of implicit polynomial surfaces and curves requires some understanding of them, how to fit them to data, and the instabilities that sometimes arise in the zero sets of the fitted polynomials. An excellent introduction to these polynomials and one approach to fitting to data is given in [23]. In this paper, we address the problem of instabilities in the fitted curves and surfaces.

Consider the following example to illustrate the instability problem. A data set is shown in Fig. 1a. In Fig. 1c, a slight variation of the data set is shown, specifically, missing a few points at one end of the curve. The third degree polynomial fits to the data sets are shown in Fig. 1b and 1d (Note that the scales in Fig. 1b and 1d are different than the ones in Fig. 1a and 1c). The scales in Fig. 1b and 1c are chosen to be bigger in order to give a feeling for how the polynomials look globally. The polynomials fit the data sets very well. The vector of coefficients for the polynomials (organized from left to right as the coefficient of the constant, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, and y^3, respectively) are :

Polynomial in Fig. 1b:

$$(-137, -2.3, 4.5, 0.03, 0.2, -0.2, 0, 0, 0, 0.001)$$

Polynomial in Fig. 1d:

$$(-42, -9.3, 5.7, 0.2, 0.2, -0.3, -0.001, 0, 0.0, 0.003)$$

Unfortunately, the coefficients of the two polynomials differ greatly, and the two curves also differ greatly over almost all of the extent shown. A quandary arises here. On the one hand, the two curves are essentially identical over the region occupied by the data. On the other hand, the curves go off to infinity, and are the same over only a negligible portion of their extents. And, as expected, the coefficients for the two curves differ considerably since they are global descriptors of the curves. Since the curve coefficients represent the curves globally, and the curves differ greatly globally, the curves cannot be compared over the local region of interest based on their coefficients. The reason for this variability is due to the fact that the data used in fitting the polynomials provides constraints among the coefficients of a fitted polynomial, but are insufficient to uniquely determine the coefficients. In such a case, the coefficients of the fitted polynomial are roughly constrained to lie on some manifold of dimensionality lower than the number of coefficients. This paper deals with solving this problem by formulating it within a probabilistic framework in the spirit of [2]. (Another way to deal with this problem is to constrain the coefficients in some way. This is done in [12] where the coefficients are constrained to be those for a polynomial having bounded zero set.)

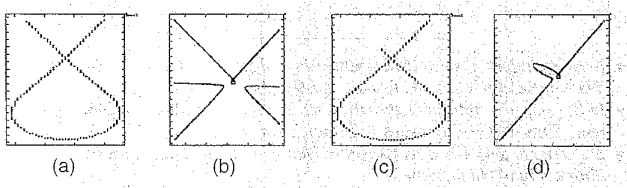


Fig. 1. (a) shows a 2D data set; (c) is a small variation of the data set in (a); (b), and (d) show the third degree polynomial fits to the data sets in (a) and (c), respectively. The scales in (b) and (c) are chosen to be bigger in order to give a feeling for how the polynomials look globally.

Another related problem that needs to be addressed is that polynomial models represent whole objects or large patches of objects. Even if the coefficients of a polynomial can be estimated accurately from data over the entire patch, a rangefinder in fixed position looking at a 3D surface sees just a portion of the surface, thus often not providing enough data to estimate the polynomial coefficients accurately. Then, how should the rangefinder move in order to collect data along additional portions of the surface so as to reduce the uncertainty in the estimates of the polynomial coefficients as quickly as possible? For dealing with these and other problems, the following results are developed in the paper.

3 Summary of Results

Let $\mathbf{Z} = \{Z_1, Z_2, \ldots, Z_N\}$ be a set of noisy data points along a curve or over a patch of the polynomial object surface. Let α be the vector of coefficients for an implicit polynomial. In Section 5, a model is proposed for the probability $p(\mathbf{Z}|\alpha)$, i.e., $p(Z_1, \ldots, Z_N|\alpha)$. Using this model, the following results are derived:

In Section 5.1, it is shown that for N large, under weak conditions,

$$p(\mathbf{Z}\,|\alpha) \approx p(\mathbf{Z}|\hat{\alpha}_N)\exp\left\{-\frac{1}{2}(\alpha-\hat{\alpha}_N)^t\,\Psi_N(\mathbf{Z})(\alpha-\hat{\alpha}_N)\right\}$$

where $\hat{\alpha}_N$ is the maximum likelihood estimate (MLE) of α based on the data, $\mathbf{Z}$, and $\Psi_N(\mathbf{Z})$ is a $d \times d$ matrix that depends on the data set, (where d is the dimension of the vector α). $\Psi_N(\mathbf{Z})$ increases with N, and (1) becomes an impulse in α as $N \to \infty$. This result turns out to be useful for exploring many of the seeming instability problems associated with polynomial coefficient estimation. Given a prior probability density function (pdf), $p(\alpha)$ for α, the a posteriori pdf, $p(\alpha\,|\mathbf{Z})$ for α given the data, is given approximately by

$$constant \times \exp\left\{-\frac{1}{2}(\alpha-\hat{\alpha}_N)^t\,\Psi_N(\mathbf{Z})(\alpha-\hat{\alpha}_N)\right\} p(\alpha).$$

If $p(\alpha)$ is broad, then $p(\alpha\,|\mathbf{Z})$ is determined largely by $p(\mathbf{Z}|\alpha)$. Hence, if the elements of $\hat{\alpha}_N$ can vary wildly with small change in the data set $\mathbf{Z}$, it is because Z does not rigidly constrain α and $p(\alpha\,|\mathbf{Z})$ is very broad in some subspace or manifold in the d-dimensional α space. The question of the intrinsic dimensionality of the coefficients of a polynomial and the subspace or manifold in which they lie is discussed in Section 5.4.

If $p(\mathbf{Z}|\alpha)$ and $p(\alpha|\mathbf{Z})$ are too broad for making reliable inferences about the object being sensed, the sensor can be moved in some direction and additional data can be taken to reduce the uncertainty. In Section 9, optimal differential sensor motion in order to maximize uncertainty reduction per unit sensor motion is discussed.

The approximation in (1) results in computationally attractive recognizers as shown in Section 5.3.

If the data set is a transformed (translated and rotated, or linearly transformed) version of the object in the database, then one cannot compare the objects in coefficient space. The solution in this case is to compute Bayesian-recognizers in the space of geometric invariants. Details of this are given in Section 6.

The beauty of this result and the preceding result is that even if there is data over only a portion of an object, object recognition can still be done by fitting a polynomial and making a single computation based on its coefficient vector or vector of invariants. Multiple computations and search in order to do subset matching are unnecessary. This is illustrated in the experiments in Section 6.

The ideas presented above extend to the case where instead of dealing with individual objects, there are classes of objects where each class consists of two or more objects. This is treated in Section 8. In this section, the problem of deciding on the appropriate polynomial degree for representing a data set is discussed as one application.

4 A Distance Measure and Fitting Implicit Polynomials

The problem addressed in this section is that of fitting implicit polynomial curves and surfaces to data. The simple way to do this for a 3D surface is to minimize the function $\sum_{(x_i,y_i,z_i)\in S} f^2(x_i, y_i, z_i)$, where S is the data set consisting of N points. This, however, may result in a poor polynomial fit to the data because $f^2(x_i, y_i, z_i)$ is not a good measure of the distance from the point (x_i, y_i, z_i) to the zero set of $f(x, y, z)$, i.e, the surface modeled by $f(x, y, z)$ (see [6] for the case of quadric curves). A much better distance measure [18], [23] is the approximate distance

$$dist^2(S, f) = \sum_{(x_i,y_i,z_i)\in S} \frac{f^2(x_i, y_i, z_i)}{\left\|\nabla f(x_i, y_i, z_i)\right\|^2} \tag{2}$$

where $\|\nabla f(x_i, y_i, z_i)\|^2$ denotes the square of the norm of the gradient.

See [23] for computationally fast ways of doing this minimization. All the curve fits in this paper are obtained using Taubin's algorithm. The surface fits can be obtained in the same way. In [12], [26] an approach to fit implicit polynomials that have bounded zero sets is introduced.

5 Asymptotic Parameter Distributions, Parameter Estimation, and Bayesian Recognition

The three problems addressed in this section are:

1) Maximum likelihood estimation of polynomial parameters.
2) The a posteriori probability density function for the polynomial coefficients given measurement data.
3) Minimum probability of error recognition of object (or object class).

The input data here is a sequence of range data points Z_1, $Z_2, \ldots, Z_N$, with $Z_i = (x_i, y_i, z_i)^t$. The discussion will be in terms of range data, but the same equations apply to 2D image

data. Let α denote the vector of coefficients of the polynomial $f(x, y, z)$ that describes the given object. Given α, the range data points $Z_1, Z_2, \ldots, Z_N$ are assumed to be statistically independent, with Z_i having probability density function (pdf)

$$p(Z_i|\alpha) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}\frac{f^2(Z_i)}{\|\nabla f(Z_i)\|^2}\right] \quad (3)$$

The assumption is that Z_i is a noisy Gaussian measurement of the object boundary in the direction perpendicular to the boundary at its closest point. The mean of Z_i is the closest point, Z_0^i, on the surface, and the variance is σ^2. The exponent is the approximate squared distance discussed in Section 4. See Appendix A for a discussion of why

$$\frac{f^2(Z_i)}{\|\nabla f(Z_i)\|^2}$$

is the approximate squared distance from a point Z_i to the surface. Physical considerations leading to this model are discussed in [2]. Even though our analysis is in terms of the probabilistic model (3), our general approach is easily adapted to any probabilistic data generation model.

Using (3), the joint probability of the data points $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_N)$ is

$$p(Z_1, Z_2, \ldots, Z_N|\alpha) = \frac{1}{\sqrt{(2\pi\sigma^2)^{N/2}}} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\frac{f^2(Z_i)}{\|\nabla f(Z_i)\|^2}\right] \quad (4)$$

The maximum likelihood estimate $\hat{\alpha}_N$ of α given the data points is the value of α that maximizes (4). Maximizing this function is equivalent to minimizing the function,

$$\sum_{i=1}^{N}\frac{f^2(Z_i)}{\|\nabla f(Z_i)\|^2},$$

i.e., the maximum likelihood estimate is the same as the unconstrained least squares polynomial fit to the data.

A very useful tool for formulating the problems of object recognition and parameter estimation is an asymptotic approximation to the joint likelihood function, (4). The next section is a brief description of this approximation.

5.1 The Asymptotic Form for the Likelihood of the Data

Let $X_1, X_2, \ldots, X_N$ be N independent identically distributed random variables with common pdf $p(\mathbf{X}|\alpha)$. Let $p(\mathbf{X}|\alpha)$ be an arbitrary probability density function (pdf) that satisfies the following conditions.

1) The second partial derivatives,

$$\frac{\partial^2}{\partial\alpha_i\partial\alpha_j}p(\mathbf{X}|\alpha),$$

are continuous functions of α for almost all $\mathbf{X}$.

2) For any positive ϵ, $\lim_{N\to\infty} P[|\hat{\alpha}_N - \alpha| < \epsilon|\alpha] = 1$ where $\hat{\alpha}_N$ is the maximum likelihood estimate of α based on $X_1, X_2, \ldots, X_N$, i.e., the value of α for which

$$p(X_1, \ldots, X_N|\alpha)$$

is a maximum.

3) $E\left[\frac{\partial^2}{\partial\alpha_i\partial\alpha_j} ln\ p(\mathbf{X}|\alpha)\right]$ is bounded.

4) The second derivative matrix,

$$\frac{\partial^2}{\partial\alpha_i\partial\alpha_j} ln\ p(\mathbf{X}|\alpha)$$

is positive definite.

A statistical result [27], used recently in a more general version for vision applications [1], [2], is that the joint pdf of the samples is asymptotically Gaussian in α, i.e.,

$$p(X_1, \ldots, X_N|\alpha) = \prod_{i=1}^{N} p(X_i|\alpha)$$
$$\approx \left[p(X_1, \ldots, X_N|\hat{\alpha}_N)\right] \exp\left\{-\frac{1}{2}(\alpha - \hat{\alpha}_N)^t\, \Psi(\mathbf{X})(\alpha - \hat{\alpha}_N)\right\} \quad (5)$$

where $\hat{\alpha}_N$ is the maximum likelihood estimate (MLE) of α based on the data $X_1, \ldots, X_N$ and $\Psi(\mathbf{X})$ is the second derivative matrix having i,jth component

$$-E\left[\frac{\partial^2}{\partial\alpha_i\partial\alpha_j} ln\ p(\mathbf{X}|\alpha)\right]$$

$\Psi(\mathbf{X})$ is called the Fisher Information matrix. An approximation to this matrix is $\Psi_N(\mathbf{X})$ having i,jth component $-\sum_{k=1}^{N}\frac{\partial^2}{\partial\alpha_i\partial\alpha_j} ln\ p(X_k|\alpha)\big|_{\alpha=\hat{\alpha}_N}$. The matrix $\Psi_N(\mathbf{X})$ is referred to in this paper as the information matrix of α. Also, in the rest of the paper, $\Psi_N(\mathbf{X})$ will be denoted as Ψ_N for convenience.

A generalization of (5) to application to implicit polynomials is

$$p(Z_1, \ldots, Z_N|\alpha) \approx \left[p(Z_1, \ldots, Z_N|\hat{\alpha}_N)\right] \exp\left\{-\frac{1}{2}(\alpha - \hat{\alpha}_N)^t\, \Psi_N(\alpha - \hat{\alpha}_N)\right\} \quad (6)$$

where the Z_i are not identically distributed since each has a different mean—the closest point on the polynomial boundary. Nevertheless, this asymptotic form applies to our problem [6]. Hence, all the useful information about α is summarized in the quadratic form in the exponent of (6).

5.2 A Posteriori Distribution for the Parameters Given the Data

The a posteriori distribution of α given the data, $p(\alpha|Z_1, Z_2, \ldots, Z_N)$, is proportional to $p(Z_1, Z_2, \ldots, Z_N|\alpha)\, p(\alpha)$. Using the asymptotic approximation (6), this joint distribution can be written as follows:

$$p(Z_1, Z_2, \ldots, Z_N, \alpha) \approx p(Z_1, \ldots, Z_N|\hat{\alpha}_N) \exp\left[-\frac{1}{2}(\alpha - \hat{\alpha}_N)^t\, \Psi_N(\alpha - \hat{\alpha}_N)\right] p(\alpha) \quad (7)$$

Equation (7) is a very useful form because it tells us about the uncertainty in the parameter values given the data points. The information matrix Ψ_N defines an ellipsoid around $\hat{\alpha}_N$ in the d-dimensional coefficient space. The axes of this ellipsoid are the directions of the eigenvectors of the information matrix, and the lengths of the axes are equal to the square roots of the eigenvalues. The interior of this ellipsoid is the region containing most of the probability mass for α, i.e., it is the region in which α is likely to lie. The volume of this ellipsoid gives a measure of the uncertainty in the parameter estimates. If the volume is large, then it implies that the coefficients are not reliable. If the coefficients are not reliable, neither will be the invariants that are functions of these coefficients. Then, instead of using the existing measurements to recognize the object, the system can collect more data in order to improve the parameter estimates (i.e., reduce the uncertainty volume). Details of how to collect more data in order to reduce the uncertainty volume as quickly as possible are given in Section 9.

Whaite and Ferrie use the information matrix Ψ_N for making inferences about objects modeled as super-quadrics [29] because Ψ_N is the Hessian of their distance measure and conveys information about the uncertainty in the parameter estimates. Their approach is deterministic whereas this is probabilistic, and their "distance measure" is different than (3) and is not an approximation to a distance measure in the sense of Euclidean distance. Knowing the uncertainty in the parameter estimates, they compute the uncertainty in the spatial domain for the purpose of visual exploration. Using our probabilistic framework, the uncertainty in the spatial domain can be computed as well. However, the expressions are complicated for implicit polynomials of high degree. Also, the main objective in this paper is object recognition based on the coefficients, for which computing the uncertainty in the parameters is sufficient.

5.3 Minimum Probability of Error Object Recognition

Consider the simplest recognition scenario where there are a set of L objects, labeled by $l = 1, 2, \ldots, L$, and modeled by polynomials of the same degree n in x, y, and z. Let α_l, $l = 1, 2, \ldots, L$, denote the parameter vectors for these objects. Then, given a sequence of data points $\mathbf{Z} = Z_1, Z_2, \ldots, Z_N$ from a part of one of the objects, the minimum probability of error recognition of object type is—Choose l for which (8) is maximum

$$p(\mathbf{Z}|\alpha_l) \tag{8}$$

This requires considerable computation because the raw data $\mathbf{Z}$ is processed a total of L times in order to compute (8) for each l. Using the asymptotic approximation (6) gives a computationally attractive recognition rule.

Using (6),

$$p(\mathbf{Z}|\alpha_l) \approx p(\mathbf{Z}\,|\hat{\alpha}_N) \exp\left\{-\frac{1}{2}(\alpha_l - \hat{\alpha}_N)^t\, \Psi_N(\alpha_l - \hat{\alpha}_N)\right\} \tag{9}$$

where $\hat{\alpha}_N$ is the maximum likelihood estimate of α and is the set of parameter values of the unconstrained polynomial fit of degree n to the data. Thus, $p(\mathbf{Z}|\hat{\alpha}_N)$ is independent of l, and maximizing the function in (9) is equivalent to minimizing the quadratic form, $(\alpha_l - \hat{\alpha}_N)^t\, \Psi_N(\alpha_l - \hat{\alpha}_N)$. Thus, the recognition rule becomes—Choose l for which (10) is minimum

$$(\alpha_l - \hat{\alpha}_N)^t\, \Psi_N(\alpha_l - \hat{\alpha}_N) \tag{10}$$

Note that $\hat{\alpha}_N$ and Ψ_N constitute sufficient statistics for this recognition. Equation (10) is a low computational cost recognition rule because here the data is involved just once, to compute the information matrix, unlike in (8) where the data is used L times. This recognition rule gives approximately the same result as does (8) when the number of data points, N, is at least a few times larger than the number of polynomial coefficients. This approximate equivalence is verified experimentally in the next section.

Note that (10) is a Mahalanobis distance measure. Using this distance measure is in fact equivalent to checking how well the data set $\mathbf{Z}$ is fit by the polynomial having coefficient vector α_l. Hence, (10) is a way of computing local data-to-object distance in terms of parameter vectors which are global descriptors of the object. The Mahalanobis distance uses those directions in coefficient space that are pertinent to the local data.

The significance of the Mahalanobis distance

$$(\alpha_l - \hat{\alpha}_N)^t\, \Psi_N(\alpha_l - \hat{\alpha}_N)$$

is illustrated in Fig. 2. A data set representing the letter e is shown, along with the zero set of the best fitting third degree polynomial having coefficient vector $\hat{\alpha}_N$. Two other zero sets are shown, those for third degree polynomials having coefficient vectors α_1 and α_2, where $\alpha_1 - \hat{\alpha}_N$ and $\alpha_2 - \hat{\alpha}_N$ have the directions of the eigenvectors associated with the minimum eigenvalue and the maximum eigenvalue, respectively, of Ψ_N. Furthermore, the Euclidean distances $\|\alpha_1 - \hat{\alpha}_N\|$ and $\|\alpha_2 - \hat{\alpha}_N\|$ are the same. Hence, it can be seen that *the polynomial zero set for α_1 is very close to that for $\hat{\alpha}_N$ in the vicinity of the original data set, whereas the zero set for α_2 is quite different.* The Mahalanobis distance effectively uses a subspace of the coefficient space that is spanned by those eigenvectors of the information matrix corresponding to the eigenvalues that are larger than ϵ, where ϵ is small. Thus, the Mahalanobis distance is a metric that is useful for comparing two polynomial zero sets in the vicinity of the data set, based on the coefficient vectors.

A point to be noted is that the approximate distance measure used in (3) may be a poor approximation at a point far from the polynomial zero set. Nevertheless, (10) still seems to produce a good recognizer because (9) is accurate and large for α_l for the object producing the measured data, and the right side of (9) will be small for the other α_l, even though the right side of (9) may be a poor approximation to the left side.

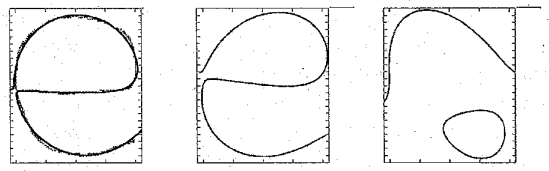
5.4 Minimum Probability of Error Recognition in Important Special Cases

This section deals with the description of special cases that arise in a recognition scenario and the computation of minimum probability of error recognizers for them.

Special case 1 arises when fitting any *unrestricted* polynomial. Consider the following example of an unrestricted second degree polynomial,

$$f(x, y) = \alpha_0 x^2 + \alpha_1 xy + \alpha_2 y^2 + \alpha_3 x + \alpha_4 y + \alpha_5,$$

to illustrate the problem. Multiplying all the coefficients of this polynomial by the same constant does not change the zero set. Hence, the likelihood of the data given α is constant along lines in α space passing through the origin. The information matrix Ψ_N in this case will be singular with five nonzero eigenvalues and one eigenvalue that is identically zero.



(a) (b) (c)

Fig. 2. Figure illustrating the significance of the Mahalanobis distance for doing object recognition. (a) shows the third degree polynomial fit for the character *e*. $\hat{\alpha}_N$ is the coefficient vector for this polynomial zero set; (b) shows the polynomial zero set corresponding to the coefficient vector α_1; (c) is the polynomial zero set corresponding to the coefficient vector α_2. $\alpha_1 - \hat{\alpha}_N$ and $\alpha_2 - \hat{\alpha}_N$ have the directions of the eigenvectors associated with the minimum eigenvalue and the maximum eigenvalue, respectively, of Ψ_N; and the Euclidean distances $\|\alpha_1 - \hat{\alpha}_N\|$ and $\|\alpha_2 - \hat{\alpha}_N\|$ are the same.

The solution in this case is to reparameterize the polynomial in terms of a reduced set of coefficients. A suitable parameterization, however, is not unique. One possibility is to parameterize the unit sphere centered at the origin, i.e., specify each line by its intersection with the unit sphere, which is its direction cosines. If α is d-dimensional, there are $d - 1$ independent direction cosines. A second possibility is to parameterize a hyperplane in the d-dimensional α space, in which case the reduced parameter set β is the projection of α on the hyperplane. One special case of this is to simply set one coefficient, e.g., α_0 or α_5 (for a second degree polynomial) to 1. The remaining five coefficients then determine the curve. A second special case is to choose the hyperplane at unit distance from the origin and perpendicular to $\hat{\alpha}_N$.

Having picked an appropriate parameterization, $\alpha = \mathbf{T}_\beta(\beta)$, where $\mathbf{T}_\beta(\beta)$ is the mapping from β to α, the question is how to compute the asymptotic approximation of the likelihood of the data given β so as to be able to compute minimum probability of error recognizers in the β parameter space. One approach is to compute the MLE $\hat{\beta}_N$ and the asymptotic form for $p(\mathbf{Z}|\beta)$ directly. For example, in the case of a second degree polynomial

$$f(x, y) = \alpha_0 x^2 + \alpha_1 xy + \alpha_2 y^2 + \alpha_3 x + \alpha_4 y + \alpha_5,$$

when the parameterization is through the unit sphere centered at the origin, the transformation $\mathbf{T}_\beta(\beta)$ from $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4)$ to $(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$ is given by

$$(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) \equiv (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \sqrt{1 - \rho^2}),$$

where

$$|\beta_i| \le 1,\ 0 \le i \le 4,\ \sum_{i=0}^{4} \beta_i^2 \le 1,$$

and

$$\rho^2 = \beta_0^2 + \beta_1^2 + \beta_2^2 + \beta_3^2 + \beta_4^2.$$

Hence, this second degree polynomial is now specified by five β parameters. The problem with this approach is that computing the MLE $\hat{\beta}_N$ involves a constrained polynomial fit to the data and hence is computationally more expensive than is unconstrained fitting. An alternative is to start with $\hat{\alpha}_N$, the unconstrained estimate of α, and then solve for $\hat{\beta}_N$ by projecting $\hat{\alpha}_N$ on the manifold in α space specified by β. Computational cost for the suggested parameterizations is negligible here. The information matrix Ψ_N^β is computed using the chain rule by

$$\Psi_N^\beta = D\mathbf{T}_\beta^t \Psi_N D\mathbf{T}_\beta \Big|_{\beta = \hat{\beta}_N} \tag{11}$$

where the matrix, $D\mathbf{T}_\beta$ of dimension $d \times d - 1$, is the Jacobian matrix of T_β having i, jth element $\frac{\partial \alpha_i}{\partial \beta_j}$. The asymptotic approximation for the likelihood of the data given β then is

$$p(Z_1, \ldots, Z_N|\beta) \approx \left[p(Z_1, \ldots, Z_N|\hat{\beta}_N)\right] \exp\left[-\frac{1}{2}(\beta - \hat{\beta}_N)^t \Psi_N^\beta (\beta - \hat{\beta}_N)\right] \tag{12}$$

Hence, this alternative approach computes the same asymptotic approximation as does the direct approach, but at much reduced computational cost.

Special case 2 arises when fitting a polynomial of degree greater than required. Let the data points cover the ellipse

$$\begin{aligned} a(x - x_0)^2 + b(y - y_0)^2 - 1 &= 0 \\ &= ax^2 + by^2 - 2ax_0 x - 2by_0 y + ax_0^2 + by_0^2 - 1 \\ &= 0 \end{aligned}$$

and be somewhat noisy. This data set is well approximated by an unrestricted second degree polynomial in x, y; denote this coefficient vector $\tilde{\alpha}$, and the polynomial $\tilde{f}(x, y)$. The information matrix, Ψ_N^β, in the β subspace will be nonsingular, and recognition can be carried out in this subspace. If a polynomial of degree greater than 2 (3, for example) is fit to the data, the fit will be good but Ψ_N^β may be singular. To understand the last statement, consider the third degree polynomial $[\tilde{f}(x, y)][cx + dy - 1]$. If the line $cx + dy - 1 = 0$ does not come close to the ellipse data, the sum of squared distances from the ellipse data to this third degree polynomial will be independent of c and d. Hence, Ψ_N^β will be singular. On the other hand, the third degree polynomial that fits the noisy ellipse data the best will have a sum of

squared errors very slightly smaller than that for polynomial $\tilde{f}(x, y)$. Hence, Ψ_N^{β} evaluated at the resulting $\hat{\beta}_N$ will be close to but not completely singular. One solution to object recognition in this case is to first compute the right degree of the polynomial for a given data set and then compute minimum probability of error recognizers in the β subspace for this polynomial. This is discussed briefly in Section 8; an example and additional discussion is given in [21].

Special case 3 is one that arises when the data set is insufficient to reliably estimate all the parameters of the polynomial. For example, if the data set is noisy and over only a fifth of the ellipse, then a second degree polynomial is necessary to approximate the data set. However, since the data set is over only a small portion of the ellipse, it is insufficient to reliably estimate all the parameters of the unrestricted second degree polynomial. The information matrix in the β subspace will be singular with the number of significantly nonzero eigenvalues being less than 5. The solution to doing recognition in this case is to collect additional data over more of the object surface until it is possible to reliably estimate all the coefficients. Details are given in Section 9.

5.5 Experimental Results

For all the results derived in this paper, the asymptotic Bayesian approximation (6) is used for the joint likelihood of the data points given α. Hence, the first question is how good is this approximation. Plots of the true likelihood of the data vs. the asymptotic approximation are given in [21]. In [21], the left and right sides of (6) are plotted as functions of α for two different data sets, illustrating the appropriateness of the approximation.

The first set of experiments in this section illustrate the use of the Mahalanobis distance for recognition. The data sets are shown in Figs. 3 and 4. The data sets are handwritten characters and are all well fit by third degree polynomials. Thus, these data sets illustrate the performance of the recognizer for unbounded polynomials. The data sets in Fig. 3a, 3b, 3c, and 3d are the objects in the database. These data sets correspond to the alphabets 'e', 's', 't', and 'r'. Also shown in these figures are the best third degree polynomial fits to the data. The data sets in Fig. 4a, 4b, 4c, and 4d are other instances of the handwritten characters 'e' and 'r'. All the data sets are scaled and translated to lie within a square box of size 10 centered at the origin. This is to make sure that all the objects are in roughly standard position and of the same size.

Mahalanobis distances (divided by the number of data points) from these four data sets to the objects in the database are:

	char 'e' (3a)	char 's' (3b)	char 't' (3c)	char 'r' (3d)
char 'e' (4a)	0.321	10.23	10.78	13.7
char 'e' (4b)	0.398	11.02	12.92	14.61
char 'r' (4c)	15.84	13.97	13.91	0.432
char 'r' (4d)	15.64	13.13	14.46	0.522

From the results one can see that the recognizer picks the correct letter in all the cases even though the data sets in Fig. 4 are very different compared to the corresponding ones in the database.

All experiments were run on a SPARC 2. Polynomial fitting takes 1-2 sec for fitting a fourth degree polynomial in x, y to about 200 points. The computation of the uncertainty matrix takes 1-2 sec for a fourth degree polynomial in x, y because there are explicit expressions for the elements of the matrix. Other computations incur negligible time. These computation times can be sped up by an order of magnitude using new fitting algorithms and parallelization.

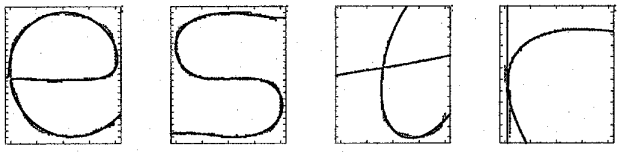


Fig. 3. Database letters 'e', 's', 't', and 'r'. Figures show the data sets and the best third degree polynomial fits.

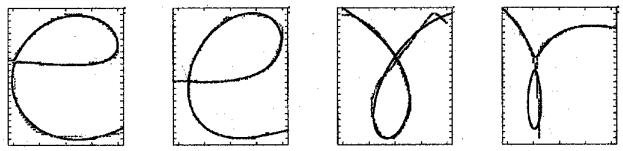


Fig. 4. Other instances of the letters 'e' and 'r'. Figures show the data sets and the best third degree polynomial fits.

6 Bayesian Recognizer Based on Geometric Invariants

In the preceding section, the solution to the simplest recognition problem was presented. The solution involved computing $p(\mathbf{Z}|\alpha_l)$ where α_l is the coefficient vector for polynomial curve or surface models for objects in class l. A computationally practical recognizer resulted from the shape of $p(\mathbf{Z}|\alpha_l)$ being Gaussian in α_l as the number of data points in $\mathbf{Z}$ becomes large. In this section, $p(\mathbf{Z}|\mathbf{G}_l)$ is examined, where $\mathbf{G}_l$ is a vector having as components the algebraic invariants for objects in class l, and a computationally practical procedure is developed for evaluating and manipulating this function. The invariants $\mathbf{G}_l$ can be viewed as the parameters specifying the data distribution for object class l. This provides the solution to optimal object recognition based on invariants when object classes may be described by distributions of different complexities and different invariants and numbers of invariants are used for different object classes.

If the data set that is modeled by the polynomial with coefficient vector α is a transformed (rotated, translated or linearly transformed) version of the data set approximated by the polynomial with coefficient vector, α', then one cannot compare coefficient vectors α and α' for determining whether the objects represented by the two polynomials are the same. This is because, even if the polynomial coefficients are stably determined by the data sets, α' will be a linear transformation of α. The solution in this case, for comparing the polynomials having coefficient vectors α and α', is to compare the sets of invariants for the two polynomials.

Invariants are functions of polynomial coefficients that do not change when the coordinate system undergoes transformations, and hence are descriptors of shape only and thus a good candidate for recognition purposes. An example of Euclidean invariants are the lengths of the major and minor axes of an ellipse. These are invariant to translation and rotation of the ellipse.

Classical work [10] presents the *symbolic method* for determining all the affine invariants of the leading form of a polynomial, i.e., the terms of highest degree. [15] contains a treatment of moment invariants. References [9], [28] contain an extensive treatment of explicit algebraic invariants for second degree polynomial curves. Taubin and Cooper in [25], [24] find many invariants that are easy to compute, as they are expressed as eigenvalues of relatively small matrices, or the coefficients of the characteristic polynomials of these matrices. Taubin's invariants can be designed to be functions of the terms of any or all degrees in the polynomial. (Taubin's formulation also applies to moment invariants.) The invariants used in this paper are explicit invariants obtained by Keren using symbolic computation [13].

Details of Keren's approach for finding explicit invariants that are polynomial functions of the coefficients of the polynomial are given in [11]. An example of the invariants that are found by this technique are the following four relative invariants of a fourth degree polynomial in x, y (a_{ij} is the coefficient of $x^i y^j$). The first two are Euclidean invariants, the last two are affine invariants. The first three invariants are functions of the coefficients of the fourth degree monomials only, whereas the last invariant is a function of all the coefficients. Since scaling the coefficients should not change the invariants, and in order to obtain the needed absolute invariants, we use ratios of these relative invariants, i.e.,

$$\left(g_1 = \frac{q_1}{q_4},\ g_2 = \frac{q_2}{q_4},\ g_3 = \frac{q_3}{q_4}\right)$$

in the experiments.

$$\begin{aligned}
q_1 &= 3a_{13}^2 - 8a_{04}a_{22} + 2a_{13}a_{31} + 3a_{31}^2 - 32a_{40}a_{04} - 8a_{22}a_{40},\\
q_2 &= 3a_{04}^2 + 2a_{04}a_{22} + a_{13}a_{31} + 2a_{04}a_{40} + 2a_{22}a_{40} + 3a_{40}^2,\\
q_3 &= a_{22}^2 - 3a_{13}a_{31} + 12a_{04}a_{40},\\
q_4 &= 144a_{40}a_{04}a_{00} - 36a_{40}a_{03}a_{01} + 12a_{40}a_{02}^2 - 36a_{31}a_{13}a_{00}\\
&\quad + 9(a_{31}a_{12}a_{01} + a_{31}a_{03}a_{10}) - 6a_{31}a_{11}a_{02} + 12a_{22}^2a_{00}\\
&\quad - 6(a_{22}a_{21}a_{01} + a_{22}a_{12}a_{10}) + 4(a_{22}a_{02}a_{20}) + 2a_{22}a_{11}^2\\
&\quad + 9(a_{13}a_{30}a_{01} + a_{13}a_{21}a_{10}) - 6a_{13}a_{20}a_{11} - 36a_{04}a_{30}a_{10}\\
&\quad + 12a_{04}a_{20}^2 - 6a_{30}a_{12}a_{02} + 9a_{30}a_{03}a_{11} + 2a_{21}^2a_{02} - a_{12}a_{21}a_{11}\\
&\quad - 6a_{21}a_{03}a_{20} + 2a_{12}^2a_{20}
\end{aligned}$$

6.1 Mahalanobis Distance Between Two Sets of Invariants

This section deals with using explicit invariants for minimum probability of error object recognition. Let $\mathbf{G} = (g_1(\alpha), g_2(\alpha), \ldots g_k(\alpha))^t$ be a set of k invariants ($k < d$), where each invariant $g_i(\alpha)$ is a polynomial function (or a ratio of polynomial functions) in the set of coefficients α. Consider the simplest case for recognition, where the database consists of a set of L objects labeled $l = 1, 2, \ldots, L$. All objects are modeled by polynomials of the same degree and each object is characterized by one point α_l in coefficient space. Let $\mathbf{G}_l$ denote the vector of invariants for object l. Then, if the object to be recognized is a transformed version of one of the database objects, from an analysis similar to the one in Section 5.3, the minimum probability of error recognition rule is—Choose l for which $p(\mathbf{Z}|\mathbf{G}_l)$ is maximum.

Thus, the problem is to compute the likelihood of the data given $\mathbf{G}$. One approach is to reparameterize the polynomial in terms of $\mathbf{G}$, and then compute $p(\mathbf{Z}|\mathbf{G})$. The problem with reparameterizing the polynomial in terms of $\mathbf{G}$, however, is that the number of invariants is smaller than the number of parameters and hence there is not a 1-1 correspondence between α and $\mathbf{G}$. This is because the maximum of number of invariants = (number of parameters)—(number of degrees of freedom of the transformation), and, in many cases, one may not be able to find all of them. Hence, in order to get a 1-1 correspondence between α and $\mathbf{G}$, a vector of additional parameters $\mathbf{H}$ is appended such that there is a 1-1 correspondence between α and the augmented vector $\Lambda^t = (\mathbf{G}^t, \mathbf{H}^t)$. The $\mathbf{H}$ parameters are referred to as nuisance parameters. The question is : how to pick these nuisance parameters? Any set of $\mathbf{H}$ parameters that gives a 1-1 correspondence between α and Λ can be used.

Under the assumption that

$$\exp\left\{-\frac{1}{2}(\alpha - \hat{\alpha}_N)^t\ \Psi_N(\alpha - \hat{\alpha}_N)\right\} \tag{13}$$

is concentrated about $\hat{\alpha}_N$, we assume that $\mathbf{G} - \hat{\mathbf{G}}_N$ is well approximated by

$$D\mathbf{G}(\alpha - \hat{\alpha}_N) \tag{14}$$

where $\frac{\partial g_i(\alpha)}{\partial \alpha}$ is a column vector, DG is the Jacobian matrix

$$\left[\frac{\partial g_1(\alpha)}{\partial \alpha}\ \frac{\partial g_2(\alpha)}{\partial \alpha} \cdots \frac{\partial g_k(\alpha)}{\partial \alpha}\right]^t$$

evaluated at $\hat{\alpha}_N$, and $\hat{\mathbf{G}}_N$ is the vector

$$\left[g_1(\hat{\alpha}_N), g_2(\hat{\alpha}_N), \ldots, g_k(\hat{\alpha}_N)\right]^t.$$

This is the representation of $\mathbf{G}$ by the constant and linear terms of a Taylor series expansion about $\hat{\mathbf{G}}_N$. Since the approximation is for a small region around $\hat{\alpha}_N$, the linear approximation might be good enough. If however, this approximation is not good, then one has to either use higher degree terms in the Taylor series expansion or use a mixture of gaussians to model the likelihood of the data given $\mathbf{G}$. Details on using a mixture of gaussians are given in [14].

In order to get a local 1:1 mapping between α and $\Lambda^t = (\mathbf{G}^t, \mathbf{H}^t)$, we can use any matrix $\mathbf{B}$ such that

$$\begin{pmatrix} \mathbf{G} - \hat{\mathbf{G}}_N \\ \mathbf{H} \end{pmatrix} = \begin{bmatrix} D\mathbf{G} \\ \mathbf{B} \end{bmatrix} (\alpha - \hat{\alpha}_N)$$

is a nonsingular transformation to get a new set of variables for representing α. Denote

$$\begin{bmatrix} D\mathbf{G} \\ \mathbf{B} \end{bmatrix}$$

by $\mathbf{C}$. Hence,

$$\alpha - \hat{\alpha}_N = \mathbf{C}^{-1}\begin{pmatrix} \mathbf{G} - \hat{\mathbf{G}}_N \\ \mathbf{H} \end{pmatrix}$$

is a local 1:1 onto linear transformation, and

$$\exp\left[-\frac{1}{2}(\alpha - \hat{\alpha}_N)^t \Psi_N (\alpha - \hat{\alpha}_N)\right]$$
$$= \exp\left[-\frac{1}{2}\left(\left(\mathbf{G} - \hat{\mathbf{G}}_N\right)^t, \mathbf{H}^t\right)\mathbf{C}^{-t}\Psi_N\mathbf{C}^{-1}\left(\left(\mathbf{G} - \hat{\mathbf{G}}_N\right)^t, \mathbf{H}^t\right)^t\right] \quad (15)$$

For N large, (15), as a function of $\begin{pmatrix} \mathbf{G} \\ \mathbf{H} \end{pmatrix}$, is highly concentrated about the point $\begin{pmatrix} \hat{\mathbf{G}}_N \\ 0 \end{pmatrix}$. We assume a uniform distribution, $p(\mathbf{H})$, for $\mathbf{H}$. Then, integrating out $\mathbf{H}$ gives

$$p(\mathbf{Z}\,|\mathbf{G}) \approx \textit{constant} \times$$
$$\int_{-\infty}^{\infty} \exp\left[-\frac{1}{2}\left(\left(\mathbf{G} - \hat{\mathbf{G}}_N\right)^t, H^t\right)\mathbf{C}^{-t}\Psi_N\mathbf{C}^{-1}\left(\left(\mathbf{G} - \hat{\mathbf{G}}_N\right)^t, \mathbf{H}^t\right)^t\right] \quad (16)$$
$$p(\mathbf{H})\,d\mathbf{H}$$
$$= \textit{constant} \times \exp\left[-\frac{1}{2}\left(\mathbf{G} - \hat{\mathbf{G}}_N\right)^t \Psi_N^{\mathbf{G}}\left(\mathbf{G} - \hat{\mathbf{G}}_N\right)\right]$$

We now look at an approach for computing $\Psi_N^{\mathbf{G}}$ in (16).

6.2 Computation of $\Psi_N^{\mathbf{G}}$

In (16), represent $\mathbf{C}^{-t}\Psi_N\mathbf{C}^{-1}$ by

$$\begin{bmatrix} \Psi_N^{\mathbf{GG}} & \Psi_N^{\mathbf{GH}} \\ \Psi_N^{\mathbf{HG}} & \Psi_N^{\mathbf{HH}} \end{bmatrix}$$

where $\left(\Psi_N^{\mathbf{GH}}\right)^t = \Psi_N^{\mathbf{HG}}$. Then, the integral (16) is

$$\textit{constant} \times \exp\left\{-\frac{1}{2}\left[(\mathbf{G} - \hat{\mathbf{G}}_N)^t \Psi_N^{\mathbf{GG}}(\mathbf{G} - \hat{\mathbf{G}}_N)\right.\right.$$
$$\left.\left. - (\mathbf{G} - \hat{\mathbf{G}}_N)^t \Psi_N^{\mathbf{HG}^t} \Psi_N^{\mathbf{HH}^{-1}} \Psi_N^{\mathbf{HG}} (\mathbf{G} - \hat{\mathbf{G}}_N)\right]\right\} \quad (17)$$

Hence,

$$\Psi_N^{\mathbf{G}} = \Psi_N^{\mathbf{GG}} - \Psi_N^{\mathbf{HG}^t}\Psi_N^{\mathbf{HH}^{-1}}\Psi_N^{\mathbf{HG}} \quad (18)$$

In general, it is impossible to structure the matrix $\mathbf{B}$ such that $\Psi_N^{\mathbf{HG}}$ is the 0 matrix. There are various ways to choose the $\mathbf{B}$ matrix. Is the integral in (16) invariant to the choice of the $\mathbf{B}$ matrix?

Since the $\mathbf{B}$ matrix is picked such that the row space of $\mathbf{C}$ is the entire Euclidean d-space, i.e., the rows of $\mathbf{C}$ are linearly independent, then if $\mathbf{B}'$ is another matrix such that the rows of

$$\mathbf{C}' = \begin{bmatrix} D\mathbf{G} \\ \mathbf{B}' \end{bmatrix}$$

are linearly independent, $\mathbf{C}' = A\mathbf{C}$ and

$$\begin{bmatrix} (\mathbf{G} - \hat{\mathbf{G}}_N) \\ \mathbf{H}' \end{bmatrix} = A\begin{bmatrix} (\mathbf{G} - \hat{\mathbf{G}}_N) \\ \mathbf{H} \end{bmatrix}.$$

Then, denoting $\Lambda' = [(\mathbf{G} - \hat{\mathbf{G}}_N)^t, (\mathbf{H}')^t]$, we see that the integral (16) transforms as

$$p(\mathbf{Z}|\mathbf{G}) \approx K \times \int_{-\infty}^{\infty} \exp -\frac{1}{2}\left[\Lambda'\mathbf{C}^{-t}\Psi_N\mathbf{C}^{-1}\Lambda'^t\right]d\mathbf{H}'$$
$$= K \times \int_{-\infty}^{\infty} \exp -\frac{1}{2}\left[\Lambda A^t A^{-t}\mathbf{C}^{-t}\Psi_N\mathbf{C}^{-1}A^{-1}A\Lambda^t\right]|A|d\mathbf{H} \quad (19)$$
$$= K \times \int_{-\infty}^{\infty} \exp -\frac{1}{2}\left[\Lambda\mathbf{C}^{-t}\Psi_N\mathbf{C}^{-1}\Lambda^t\right]d\mathbf{H}$$

where, K is a constant. Hence, the integral in (16) is invariant to the choice of the $\mathbf{B}$ matrix.

For the simplest recognition scenario, the optimum recognition rule then is—Choose l for which $p(\mathbf{Z}|\mathbf{G}_l)$ is maximum. From (16), this is equivalent to—Choose l for which the Mahalanobis distance, $(\mathbf{G}_l - \hat{\mathbf{G}}_N)^t \Psi_N^{\mathbf{G}}(\mathbf{G}_l - \hat{\mathbf{G}}_N)$ is minimum. This is because, for the simplest case, all the database objects are modeled by polynomials of the same degree and hence the only part of (16) that is a function of l is

$$\exp\left\{-\frac{1}{2}\left(\mathbf{G}_l - \hat{\mathbf{G}}_N\right)^t \Psi_N^{G}\left(\mathbf{G}_l - \hat{\mathbf{G}}_N\right)\right\}.$$

If $\Psi_N^{\mathbf{HH}}$ is singular, then $\Psi_N^{\mathbf{HH}^{-1}}$ does not exist. Then, (18) and hence $\int p(\mathbf{Z}|\Lambda)\,p(\mathbf{H})\,d\mathbf{H}$ cannot be computed. A solution in this case is to use the following approach for object recognition. Consider a database object having invariant vector $\mathbf{G}_l$. There are many coefficient vectors α that map into $\mathbf{G}_l$. Let S_l denote the set of coefficient vectors that map into $\mathbf{G}_l$. Denote $M_l = \max_{\alpha \in \mathbf{S}_l} p(\mathbf{Z}\,|\alpha)$ Then, a maximum likelihood recognition rule based on invariants is—Choose l for which M_l is maximum. $\max_{\alpha \in \mathbf{S}_l} p(\mathbf{Z}\,|\alpha)$ can always be computed approximately by using $\mathbf{G} = \mathbf{G}_l$ and finding a value of $\mathbf{H}$ for which (15) is maximum. This maximum of (15) can always be found even if $\mathbf{C}^{-t}\Psi_N\mathbf{C}^{-1}$ is singular. Details of this approach and experimental results using this approach are given in [21].

The Mahalanobis distance between two sets of invariants is a metric that is exactly invariant under Euclidean transformations of the data set, and is approximately invariant to linear transformations of the data set and hence is a good metric for comparing two data sets where one is a transformed version of the other.

In summary, object recognition using invariants for the simplest scenario is done as follows.

1) Fit the best polynomial to the data set.
2) Compute the invariants $\hat{\mathbf{G}}_N$ which are functions of the coefficients of the polynomial.
3) Compute the information matrix $\Psi_N^{\mathbf{G}}$.
4) Compute the Mahalanobis distance,

$$(\mathbf{G}_l - \hat{\mathbf{G}}_N)^t \Psi_N^{\mathbf{G}}(\mathbf{G}_l - \hat{\mathbf{G}}_N)$$

to each object in the database and pick the l for which it is a minimum.

Additional insights into the properties of and computation of $p(\mathbf{Z} \mid \mathbf{G})$ are given in [21], [14].

Experimental results illustrating the use in object recognition of the Mahalanobis distance between the sets of invariants are given in the next section. The experiments are not for large databases, but rather are to illustrate discrimination power for curves and surfaces that are similar.

7 Experimental Results

The experiments in this section illustrate the use of the Mahalanobis distance in the space of invariants for recognizing objects that are rotated and translated versions of one of the objects in the database. The simplest case of recognition is assumed where each object class in the database is a single object, i.e., a single instance.

The first set of examples to demonstrate recognition are 2D shapes. All the data sets in this experimment are handwritten characters that are well fit by fourth degree polynomials in x, y. The objects in the database are the handwritten characters, 'a', 'q', 'g', and 'w', shown in Fig. 5a, 5b, 5c, and 5d. Four relative invariants for a fourth degree polynomial in x, y (listed in Section 6) found by using our approach were used in the experiments. Since scaling the coefficients should not change the invariants, there are only three absolute invariants. One set of three absolute invariants is $g_1 = \frac{q_1}{q_4}$, $g_2 = \frac{q_2}{q_4}$, and $g_3 = \frac{q_3}{q_4}$. The values of the three invariants for the polynomial fits to the characters in the database are:

char 'a':	char 'q'	char 'g'	char 'w'
1.99	0.102	0.207	0.196
5.58	1.217	1.219	0.314
2.11	5.13	3.01	7.99

Data sets for the objects to be recognized are shown in Fig. 6a, 6b, 6c, and 6d. These are independently drawn in rotated, translated, and noisy versions of the objects in the database. The following table shows the value of the three invariants for the polynomial fit to these data sets, and the values of the Mahalanobis distances to the handwritten characters 'a', 'q', 'g', and 'w' in the database.

INV.	char 'a' (rot) (1.8, 3.9, 1.8)	char 'q' (rot) (0.1, 1.1, 5.5)	char 'g' (rot) (0.2, 1.2, 3.4)	char 'w' (rot) (0.2, 0.03, 7.5)
'a' (5a)	0.387	14.81	10.89	08.41
'q' (5b)	09.88	0.471	09.61	12.96
'g' (5c)	12.88	09.12	0.366	14.66
'w' (5d)	12.11	13.35	15.23	0.634

As seen from the results, the Mahalanobis distance is indeed small for the right object, well within the uncertainty ellipse for the correct object, and a factor of 20 smaller than that for the next closest object. Thus, the Mahalanobis distance in the space of invariants is a useful tool for comparing two data sets where one is a transformed version of the other.

The next set of experiments illustrate the performance of the recognizer for four 3D objects that differ little from one another. Fig. 7a shows the four keyboard mice used in this experiment. Fig. 7b, 7c, 7d, and 7e are the data sets and the fourth degree polynomial fits for the mice in standard position. The data sets were obtained using the Brown and Sharpe Microval Manual coordinate measuring machine.

Fig. 8a, 8b, 8c, and 8d are the data sets and fourth degree polynomial fits for the rotated and translated versions of the mice in the database. These are generated by rotating and translating the data sets for the database objects. Using our approach, seven invariants for a fourth degree polynomial in x, y, z are found [11]. These are invariant to 3D transformations of the object. The goal in this experiment is to recognize the mice in Fig. 8a, 8b, 8c, and 8d using the Mahalanobis distance and comparing the results with that using the Euclidean distance.

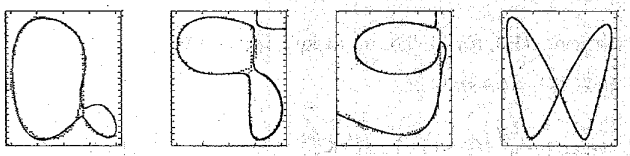


Fig. 5. Database letters 'a', 'q', 'g', and 'w'. The letters are well fit by fourth degree polynomials. Figures show the data sets and the polynomial fit.

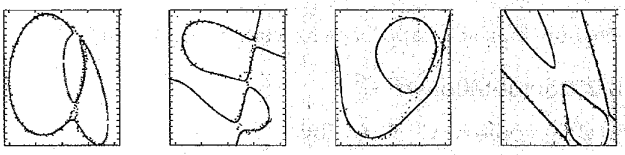


Fig. 6. Other instances of handwritten letters 'a', 'q', 'g', and 'w'. Each is a rotated, translated, and noisy version of one of those in the database. Figures show the data sets and the best fourth degree polynomial fit.

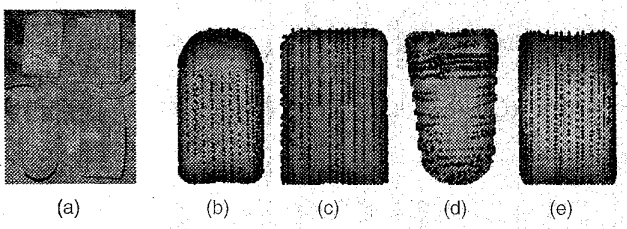


Fig. 7. (a) is the image of the mice used in the experiment. (b), (c), (d), and (e) show the data sets and the polynomial fits for the mice in standard position.

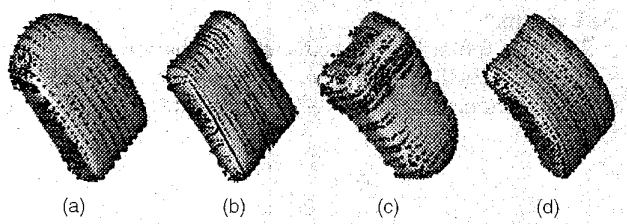


Fig. 8. Data sets and the polynomial fits for rotated and translated versions of the mice in the database.

Tables 1 and 2 show the Mahalanobis distances and the Euclidean distances, respectively, between the vector of invariants for the polynomial fits to the rotated mice and the vectors of invariants for the four mice in the database.

Thus, the Mahalanobis distance measure is a reliable measure for discriminating the right object from the rest, and has much better discriminatory power than does the Euclidean distance.

TABLE 1
MAHALANOBIS DISTANCE

	Mouse1 (rotated)	Mouse2 (rotated)	Mouse3 (rotated)	Mouse4 (rotated)
Mouse1	0.456	09.89	12.79	14.51
Mouse2	12.63	0.314	13.87	11.99
Mouse3	16.22	10.17	0.403	13.39
Mouse4	08.91	10.61	12.33	0.504

TABLE 2
EUCLIDEAN DISTANCE

	Mouse1 (rotated)	Mouse2 (rotated)	Mouse3 (rotated)	Mouse4 (rotated)
Mouse1	1.00	2.54	9.614	2.741
Mouse2	1.82	1.00	1.125	1.547
Mouse3	13.5	2.32	1.00	7.568
Mouse4	1.45	1.33	24.35	1.00

7.1 Recognizing Occluded Objects

This section deals with recognizing occluded objects.

The first set of experiments illustrate the use of the Mahalanobis distance for recognizing partially occluded 2D objects that are rotated and translated versions of one of the objects in the database. The database is the one shown in Fig. 5. Fig. 9a and 9c are the data sets for the occluded objects to be recognized. The objects in this database are modeled by fourth degree polynomials in x, y. Hence, fourth degree polynomials are fit to the occluded objects in order to compare their invariants with those for the unoccluded objects. Fig. 9b and 9d are the fourth degree polynomial fits to the data sets in 9a and 9c, respectively.

The data set in 9a is the same as the one in 5a, except that a strip of data in the center is missing. The polynomial fit nevertheless is the same as the one in 5a, thus showing that implicit functions do a good job of interpolating missing data. The data set in Fig. 9c is an occluded version of the object in 5d, where about 30% of the object is seen. The polynomial fit in this case is very different from the one in 5d.

The set of three invariants for the polynomial fit to the data set and the Mahalanobis distances (in the space of invariants) to the letter 'a', 'q', 'g', and 'w', in Fig. 5 are

INV.	char 'a' (rotated and occluded) (1.8, 3.89, 1.74)	char 'w' (rotated and occluded) (0.23, 0.29, 6.9)
char 'a'	0.399	03.98
char 'q'	14.90	04.50
char 'g'	15.50	13.17
char 'w'	11.10	0.422

For the data set in 9a, the distance to 'a' is much smaller than that to the other letters. This happens because even though the object is occluded, enouogh of the object is seen to reliable distinguish it. For the dataset in Fig. 9c, the distance to 'w' is minimum. However, the distance is small to 'a' and 'q'. This is because, the data set in 9c also fits the model for 'a' and 'q' as shown in 9e and 9f. These distance would probably be larger if strictly Euclidean invariants were used; in these experiments, one of the three invariants, q_3, is affine. An affine transformation can transform Fig. 9c to make it fit the 'a' and the 'q' fairly well. The distance to 'g' is large because the data set doesn't fit the model for 'g'.

The next set of experiments illustrate the use of the Mahalanobis distance in the space of invariants for recognizing occluded 3D objects. Fig. 10 shows the partial data (with the polynomial fit superimposed) for the mouse in Fig. 8a. The partial data in this experiment is what a stereo sensor would see when looking at the mouse from a point near the bottom left corner. The Mahalanobis distance and the Euclidean distances between the vector of invariants for the polynomial fit to the occluded object and the stored vectors of invariants are:

Distance	Mouse1:	Mouse2:	Mouse3:	Mouse4
Mahalanobis	0.604	11.23	12.03	0.842
Euclidean	1.000	1.839	1.619	0.9012

The distance to Mouse1 is the smallest. However, the distance to Mouse4 is almost the same as that to Mouse1. This is because the occluded data does not contain the curved front part of Mouse1 and since that is the part that really distinguishes Mouse1 from Mouse4, it is hard to distinguish between them based on the partial data. The distances to Mouse2 and Mouse3 are big compared to those to Mouse1 and Mouse4. The Euclidean distance does not give good recognition results with partial data. In fact, the Euclidean distance from the occluded object to Mouse4 is smaller than that to Mouse1.

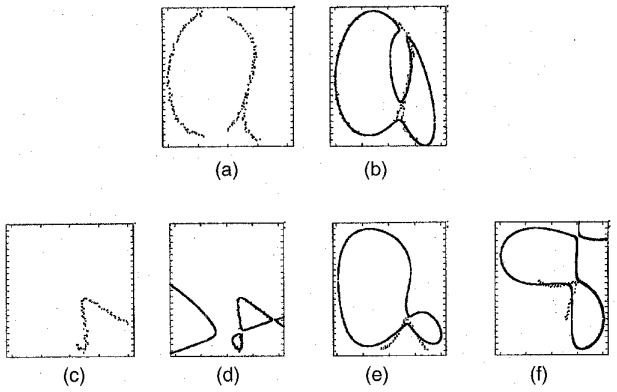


Fig. 9. Recognition of objects based on only partial data due to occlusion or poor segmentation.

8 Minimum Probability of Error Object Recognition for More Complicated Scenarios

The most complex recognition is where all the objects in class l are modeled as polynomials of the same degree n_l,

and $n_1, n_2, \ldots, n_L$ may all be different. There is an a priori distribution for the coefficient vectors for each class and for the n_l. The minimum probability of error recognizer for this scenario is—Choose l for which $p(Z_1, Z_2, \ldots, Z_N \mid$ data is from *object class l*$)p($*object class l*$)$ is maximum. The asymptotic approximation introduced in Section 5.1 can be used to compute this integral in closed form for large N, which leads to a computationally attractive recognition rule.

Details of these and experimental results are presented in [21].

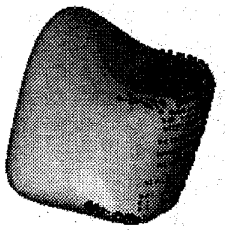


Fig. 10. Data set and polynomial fit to partial data from mouse1.

9 Where to Look Next

$p(\alpha \mid \mathbf{Z})$ may be such that the uncertainty in α is too large in order to reliably recognize the object being sensed. The vision system must then move to accumulate additional data in order to reduce this uncertainty. The problem of interest in this section is how should the sensor move in order to reduce the uncertainty as quickly as possible. Though this problem can be formulated in various ways, we assume that the sensor may move a large distance but that this is by a sequence of locally optimum moves.

The sensor could be a robot touch sensor moving a short distance along the object surface, or could be a laser rangefinder moving through a small change in position. At an instant of time, the laser rangefinder produces a depth map over a $K \times K$ grid. One situation is where the whole object is seen so that some points in the depth map are in the vicinity of the self-occluding points on the 3D surface. This data does not fit the model in this paper for two reasons. First, the depth measurement will be very noisy because the laser beam hits the 3D surface at an angle close to 90 degrees with the surface normal; second, the direction of the laser beam provides a great deal of shape information which we do not consider in the approach of this paper. Hence, we restrict consideration to the situation where the object is only partially in the range finder field of view, and a small change in sensor position produces a few returns from nearby previously unsensed regions on the object surface. For simplicity of analysis, we assume one new data point, Z_{n+1}, is produced in the new sensor position, but our analysis can directly incorporate any finite number of new data points.

Sensor motion is as follows (see Fig. 11). Consider the normal through data point Z_n to the estimated surface having coefficient vector, $\hat{\alpha}_N$. The normal intersects the estimated surface at point Z_N^*. Now consider a circle of radius δ about Z_N^* and lying in the tangential plane. The sensor is to move to look for the surface point in the vicinity of a point on the circle. In the case of the laser rangefinder, the sensor moves such that the beam lies along a line through some point Z_N^{**} on the circle and parallel to the estimated surface normal at Z_N^*. If Z_n lies very close to a singular point of the polynomial having coefficient vector $\hat{\alpha}_N$, such as occurs if there is a discontinuity in the direction of the surface normal, then the point Z_N^{**} that we seek is one that lies on the intersection of a sphere of radius δ about Z_N^* with the estimated surface based on $\hat{\alpha}_N$, and the line used is that which is perpendicular to the estimated surface at Z_N^{**}. The question of interest is how to choose Z_N^{**}. Let $\mathbf{v}_{n+1}$ be the vector of parameters that specify the line through Z_N^{**}.

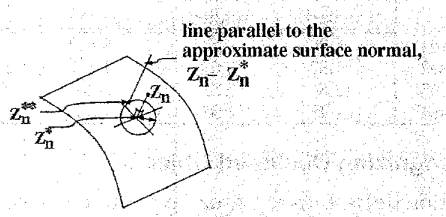


Fig. 11. Computation of the new sensor position.

The choice of $\mathbf{v}_{n+1}$ is determined by its affect on the covariance matrix for $p(\alpha \mid Z_1, \ldots, Z_{n+1})$. Specifically,

$$p(\alpha \mid Z_1, \ldots, Z_{n+1}) = constant \times p(Z_1, \ldots, Z_{n+1} \mid \alpha) p(\alpha)$$

But

$$p(Z_1, \ldots, Z_{n+1} \mid \alpha) = p(Z_{n+1} \mid Z_1, \ldots, Z_n, \alpha)\, p(Z_1, \ldots, Z_n \mid \alpha)$$

The variable Z_{n+1} depends on $Z_1, \ldots, Z_n$ only because $\mathbf{v}_{n+1}$ depends on $Z_1, \ldots, Z_n$. Hence,

$$\begin{aligned} &ln\, p(Z_1, \ldots, Z_{n+1} \mid \alpha) = \\ &ln\, p(Z_{n+1} \mid \mathbf{v}_{n+1}, \alpha) + ln\, p(Z_1, \ldots, Z_n \mid \alpha) \\ &\approx ln\, p(Z_{n+1} \mid \mathbf{v}_{n+1}, \alpha) + ln\, p(Z_1, \ldots, Z_n \mid \hat{\alpha} \\ &\quad - \frac{1}{2}(\alpha - \hat{\alpha}_n)^t \Psi_n (\alpha - \hat{\alpha}_n) \end{aligned} \tag{20}$$

From (20) we can solve for $\hat{\alpha}_{n+1}$ and Ψ_{n+1}. Upon using

$$\begin{aligned} ln\, p(Z_{n+1} \mid \mathbf{v}_{n+1}, \alpha) &\approx ln\, p(Z_{n+1} \mid \mathbf{v}_{n+1}, \hat{\alpha}_n) \\ &+ \left[\frac{\partial}{\partial \alpha} ln\, p(Z_{n+1} \mid \mathbf{v}_{n+1}, \hat{\alpha}_n)\right]^t (\alpha - \hat{\alpha}_n), \end{aligned}$$

there results

$$\hat{\alpha}_{n+1} \approx \hat{\alpha}_n + \left[\frac{\partial}{\partial \alpha} ln\, p(Z_{n+1} \mid \mathbf{v}_{n+1}, \hat{\alpha}_n)\right]^t \Psi_n^{-1} \tag{21}$$

Then, from (20) and (21), Ψ_{n+1} has the i,jth element

$$-\frac{\partial^2}{\partial_{\alpha_i}\partial_{\alpha_j}} ln\ p(Z_1, \ldots, Z_{n+1}|\hat{\alpha}_{n+1}) \approx$$

$$-\frac{\partial^2}{\partial\alpha_i\partial\alpha_j} ln\ p(Z_{n+1}|\mathbf{v}_{n+1}, \hat{\alpha}_{n+1}) - \frac{\partial^2}{\partial\alpha_i\partial\alpha_j} ln\ p(Z_1, \ldots, Z_n|\hat{\alpha}_n)$$

Hence,

$$\Psi_{n+1} \approx \Psi_n + \Psi_{\Delta,n+1} \tag{22}$$

where $\Psi_{\Delta,n+1}$ has i,jth element

$$-\frac{\partial^2}{\partial\alpha_i\partial\alpha_j} ln\ p(Z_{n+1}|\mathbf{v}_{n+1}, \hat{\alpha}_{n+1}) \approx -\frac{\partial^2}{\partial\alpha_i\partial\alpha_j} ln\ p(Z_{n+1}|\mathbf{v}_{n+1}, \hat{\alpha}_n) \tag{23}$$

What model should be used for $p(Z_{n+1}|\mathbf{v}_{n+1}, \alpha)$? The appropriate model here is different than in the preceding sections of the paper because of the dependence on $\mathbf{v}_{n+1}$. Our model here is that Z_{n+1} lies along the line determined by $\mathbf{v}_{n+1}$. Since the line is estimated to be *roughly* perpendicular to the surface being sensed, when then using Z_{n+1} to make inferences about α we use (3), the model used in previous sections. Hence, the choice of v_{n+1} influences the data point Z_{n+1} that will be observed, but once Z_{n+1} is observed, it is used as in (4) for making inferences about α and is treated as though independent of $Z_1, \ldots, Z_n$.

Finally, since Z_{n+1} is not known at the time $\mathbf{v}_{n+1}$ is chosen, the choice of $\mathbf{v}_{n+1}$ must be based on the Z_{n+1} that can occur given $\mathbf{v}_{n+1}$. Our criterion for this choice is as follows. For each $\mathbf{v}_{n+1}$, consider all possible Z_{n+1} lying along the line. For each Z_{n+1}, determine the resulting Ψ_{n+1} in accordance with (22). Choose that Z_{n+1} for which the associated $|\Psi_{n+1}|$, the determinant of Ψ_{n+1}, is a maximum. That is the v_{n+1} we use. Note, maximizing $|\Psi_{n+1}|$ is equivalent to minimizing $|\Psi_{n+1}^{-1}|$. In summary, we sense the 3D surface at a point roughly a distance δ from the last data point chosen such that the decrease in the resulting uncertainty volume for α will be as large as possible.

A reasonable alternative criterion for chosing v_{n+1} is to choose v_{n+1} to minimize

$$\int |\Psi_{n+1}^{-1}(Z_{n+1})|\, p(Z_{n+1}|\mathbf{v}_{n+1}, \hat{\alpha}_n, \Psi_n)\, dZ_{n+1}$$

The matrix $\Psi_{n+1}(Z_{n+1})$ is the Ψ_{n+1} computed in (15), and $p(Z_{n+1}|\mathbf{v}_{n+1}, \hat{\alpha}_n, \Psi_n)$ is a Gaussian distribution for Z_{n+1} along the line specified by $\mathbf{v}_{n+1}$, and which can be easily computed in terms of the intersection of the line specified by $\mathbf{v}_{n+1}$ with the surface having coefficient vector α, where α has probability distribution $p(\alpha|Z_1, \ldots, Z_n)$. The integral is an average uncertainty volume for α given $\mathbf{v}_{n+1}$.

One last possible criterion is to look at the conditional variance of Z_{n+1} given $\mathbf{v}_{n+1}$, $\hat{\alpha}_n$, and Ψ_n, i.e., the variance associated with $p(Z_{n+1}|\mathbf{v}_{n+1}, \hat{\alpha}_n, \Psi_n)$ in the preceding paragraph. There is a closed form approximation for this, so it is a low cost computation. The chosen $\mathbf{v}_{n+1}$ would then be that for which the conditional variance is a maximum. The thought is to collect a new data point where the a priori uncertainty of surface location is a maximum, thereby collecting data that can reduce uncertainty of surface location as much as possible. This concept is related to that of [2], and more recently to that of [29], but the probabilistic approach and criterion are entirely different than that of Whaite and Ferrie.

An experimental result illustrating this approach is shown in Fig. 15. The surface to be sensed is a sphere of unit radius centered at the origin. In the experiment, the system does not know that the surface is a sphere. It assumes that it is a general quadric surface. If the surface is known to be a sphere, then it can be estimated simply by taking data points from a few orthogonal views and the problem of 'where to look next' can be solved very easily. The harder problem which is addressed in this section is one where no prior knowledge of the quadric-surface parameter values is assumed.

Fig. 12a shows the sphere of unit radius centered at the origin, and the first n (n = 27) (noisy) data points. The noise variance is 0.02. The data points lie in the region

$$-0.04 \le x \le 0.04,\ 0.2 \le y \le 0.3,\ 0.9531 \le z \le 0.9798.$$

The nth data point is (0.04, 0.2, 0.97897).

The criterion that we use to get $\mathbf{v}_{n+1}$ is – Pick the $\mathbf{v}_{n+1}$ that maximizes $|\Psi_{n+1}|$, or, equivalently, minimizes $|\Psi_{n+1}^{-1}|$. Having obtained Z_{n+1}, we treat it as though independent of Z_1, Z_2, ..., Z_n. We then repeat the procedure to estimate Z_{n+2}, and so on. The next question is when to stop collecting data. There are different stopping criteria that can be used. One criterion is to stop collecting data when the variance of the coefficients become smaller than a certain amount. Depending on the application, one can use other criteria as well such as the size of the uncertainty volume $|\Psi_n^{-1}| = |\Psi_n|^{-1}$. In the experiment shown, we stop collecting data when the variance of the coefficients becomes less than 0.02. Fig. 12b shows the complete trajectory traced out by the algorithm. The trajectory spirals outward, which is expected in this example because of the symmetry in the surface. The sampling interval could have been increased in this problem in order to decrease the number of data points required.

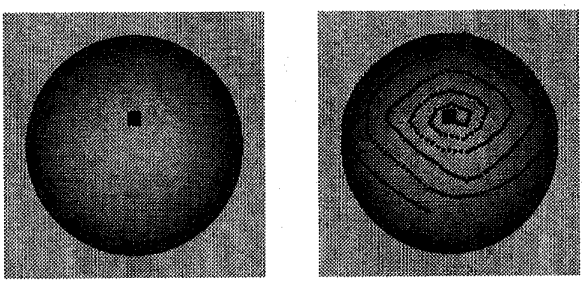


Fig. 12. (a) An a priori unknown second degree polynomial surface being sensed; (b) The sequence of data points collected by the 'Where to Look Next' algorithm.

10 Robustness and Other Desirable Properties

Recognizers based on implicit polynomial 2D curves and 3D surfaces have at least two important advantages over most other approaches. First is the use of algebraic invariants which reduces the computational cost greatly. Second is robustness to missing data.

The recognizer appears to be quite robust to considerable missing data if recognition is based on coefficients (5.3) or based on invariants for object shapes that are not complex (6.1). It is sometimes less robust for recognizing complex shapes based on invariants because invariants are highly nonlinear funcitons of the coefficients and the recognizer involves linearizations. An effective procedure then is to store two or more vectors of invariants for each object in the database of prototypes for use in our Mahalanobis recognizers. Due to space limitations in this paper, experiments providing some insight into the extent of this robustness are described in [21].

Appendix A

Appendix A deals with computing the approximate squared distance from a point to a surface.

Fig. 13 shows the distance of a point Z_i to a surface. The approximate value of the square of this distance $\left\|Z_0^i - Z_i\right\|^2$ is

$$\frac{f^2(Z_i)}{\left\|\nabla f(Z_i)\right\|^2}.$$

This can be seen from use of the first two terms of the Taylor series expansion for $f(Z)$ in the neighborhood of Z_i, where Z is a point on the surface.

$$0 = f(Z) \approx f(Z_i) + [\nabla f(Z_i)]^t (Z - Z_i) \tag{24}$$

If Z_i is close to the surface, $\nabla f(Z_i)$ will be approximately the same as $\nabla f(Z_0^i)$, where Z_0^i is the closest point on the surface to Z_i. Hence, $Z_0^i - Z_i$ has roughly the same direction as $\nabla f(Z_i)$, and thus

$$\left\|Z_0^i - Z_i\right\|^2 \approx \frac{\left[[\nabla f(Z_i)]^t (Z_0^i - Z_i)\right]^2}{\left\|\nabla f(Z_i)\right\|^2} = \frac{f^2(Z_i)}{\left\|\nabla f(Z_i)\right\|^2} \tag{25}$$

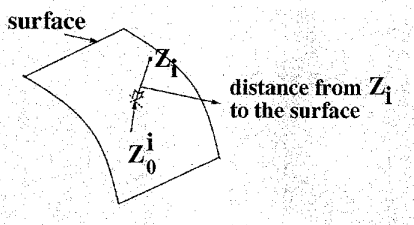


Fig. 13. Distance from a point to a surface.

Acknowledgments

This work was partially supported by the National Science Foundation under Grant No. IRI-9224963, National Science Foundation-DARPA under Grant No. IRI-8905436, and ARPA under Grant No. F49620-93-1-0501ARPA. The authors wish to acknowledge consultation with Dr. Stuart Geman and Dr. Michael Werman on a basic question concerning Bayesian recognition based on algebraic invariants. This and certain other valuable work was accomplished in the stimulating environment of The Newton Institute For The Mathematical Sciences, Cambridge University, England, during the fall of 1993.

References

[1] R. Bolle and D. Cooper, "Bayesian Recognition of Local 3D Shape by Approximating Image Intensity Functions with Quadric Polynomials," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 4, pp. 418-429, July 1984.

[2] R. Bolle and D. Cooper, "On Optimally Combining Pieces of Information , with Applications to Estimatng 3D Complex-Object Position from Range Data," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 8, no. 5, pp. 619-638, Sept. 1986.

[3] R. Bolles and P. Horaud, "3DPO: A Three-Dimensional Part Orientation System," *The Int'l J. of Robotics Research*, vol. 5, no. 3, pp. 3-26, 1986.

[4] B. Cernuschi-Frias, "Orientation and Location Parameter Estimatin of Quadric Surfaces in 3D from a Sequence of Images," PhD thesis, Brown Univ., Providence, R.I., May 1984.

[5] C. Chen and A. Kak, "A Robot Vision System for Recognizing 3D Objects in Low-Order Polynomial Time," *IEEE Trans. On Systems, Man, and Cybernetics*," vol. 19, no. 6, pp. 1,535-1,563, Nov./Dec. 1989.

[6] D. Cooper, "When Should a Learning Machine Ask for Help," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, July 1974.

[7] D. Cooper and N. Yalabik, "On the Computational Cost of Approximating and Recognizing Noise-Perturbed Straight Lines and Quadratic Arcs in the Plane," *IEEE Trans. On Computers*, vol. 25, no. 10, pp. 1,020-1032, Oct. 1976.

[8] D.B. Cooper and F. Sung, "Multiple-Window Parallel Adaptive Boundary Finding in Computer Vision," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 5, pp. 299-316, May 1983.

[9] D. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell, "Invariant Descriptors for 3-D Object Recognition and Pose," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 13, pp. 971-991, Oct. 1991.

[10] J. Grace and A. Young, *The Algebra of Invariants*. Cambridge Univ. Press, 1903.

[11] D. Keren, "Some New Invariants in Computer Vision," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, pp. 1,143-1,149, Nov. 1994.

[12] D. Keren, D.B. Cooper, and J. Subrahmonia, "Describing Complicated Objects by Implicit Polynomials," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 16, pp. 38-54, Jan. 1994.

[13] D. Keren, J. Subrahmonia, and D.B. Cooper, "Robust Object Recognition based on Implicit Algebraic Curves and Surfaces," *IEEE Conf. On Computer Vision and Pattern Recognition*, pp. 791-794, Champaign, Ill., June 1992.

[14] Z. Lei, D. Keren, and D.B. Cooper, "Recognition of Complex Free-Form Objects Based on Mutual Algebraic Invariants for Paris of Patches of Data," Tech. Report 140, LEMS, Division of Engineering, Brown Univ., Providence, R.I., Jan. 1995.

[15] V. Markandey and R.J.P. deFigueiredo, "Robot Sensing Techniques Based on High-Dimensional Movement Invariants and Tensors," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 8, pp. 186-195, Apr. 1992.

[16] O.D. Faugeras, M. Hebert, and E. Pauchon, "Segmentation of Range Data into Planar and quadratic Patches," *Proc. IEEE Conf. On Computer Vision and Pattern Recognition*, pp. 8-13, Washington, D.C., June 1983.

[17] V. Pratt, "Direct Least Squares Fitting of Algebraic Surfaces," *Computer Graphics*, vol. 21, pp. 145-152, July 1987.

[18] P. Sampson, "Fitting Conic Sections to Very Scattered Data: An Iterative Improvement of the Bookstein Algorithm," *Computer Vision, Graphics, and Image Processing*, vol. 18, pp. 97-108, 1982.

[19] K. Siddiqi and B.B. Kimia, "Parts of Visual Form: Computational Aspects," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 239-251, Mar. 1995.
[20] J. Silverman and D. Cooper, "Bayesian Clustering for Unsupervised Estimation of Surface and Texture Models.," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 10, pp. 482-495, July 1988.
[21] J. Subrahmonia, D.B. Cooper, and D. Keren, "Practical Reliable Bayesian Recognition of 2D and 3D Objects Usng Implicit Polynomials and Algebraic Invariants," Tech. Report LEMS-107, Brown Univ., Providence, R.I., May 1992.
[22] J. Subrahmonia, D. Keren, and D.B. Cooper, "Bayesian Methods for the Use of Implicit Polynomials andAlgebraic Invariants in Practical Computer Vision," *Proc. SPIE Conf.*, Boston, Mass., June 1992.
[23] G. Taubin, "Estimation of Planar Curves, Surfaces and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 13, pp. 1,115-1,138, Nov. 1991.
[24] G. Taubin and D.B. Cooper, "3D Object Recognition and Positioning with Algebraic Invariants and Covariants," pp. 147-182, July 1990. A chapter in *Symbolic and Numerical Computations Towards Integration*, pp. 147-182. B.R. Donald, D. Kapur, and J. Mundy, eds., Academic Press, 1992.
[25] G. Taubin and D.B. Cooper, "2D and 3D Object Recognition and Positioning System Based on Moment Invariants," *Geometric Invariance in Machine Vision*, J. Mundy and A. Zisserman, eds., pp. 375-397, 1992.
[26] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D.J. Kriegman, "Parameterized Families of Polynomials for Bounded Algebraic Curve and Surface Fitting," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 16, pp. 287-304, Mar. 1994.
[27] A. Wald, "Tests of Statistical Hypotheses Concerning Several Parameters when the Number of Observations is Large," *Trans. Of American Math. Society*, vol. 54, pp. 426-482, 1948.
[28] I. Weiss, P. Meer, and S.M. Dunn, "Robustness of Algebraic Invariants," *First DARPA-ESPIRIT Workshop on Invariants*, pp. 345-359, Rikjavik, Iceland, Mar. 1991.
[29] P. Whaite and F.P. Ferrie, "From Uncertainty to Visual Exploration," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 13, pp. 1,038-1,049, Oct. 1991.